

6(a)	1 mark for each correct description, max 2 marks <table border="1" data-bbox="368 185 1385 450"> <thead> <tr> <th data-bbox="368 185 544 248">Register</th><th data-bbox="544 185 1385 248">Role</th></tr> </thead> <tbody> <tr> <td data-bbox="368 248 544 383">ACC</td><td data-bbox="544 248 1385 383">stores the intermediate results of arithmetic and logical operations // holds the result of a calculation</td></tr> <tr> <td data-bbox="368 383 544 450">CIR</td><td data-bbox="544 383 1385 450">holds the instruction currently being decoded and/or executed</td></tr> </tbody> </table>	Register	Role	ACC	stores the intermediate results of arithmetic and logical operations // holds the result of a calculation	CIR	holds the instruction currently being decoded and/or executed
Register	Role						
ACC	stores the intermediate results of arithmetic and logical operations // holds the result of a calculation						
CIR	holds the instruction currently being decoded and/or executed						
6(b)	1 mark per bullet point, max 2 marks e.g. • Latency may be increased • ... because the cores must communicate with one another • There is a potential for dead-lock situations • ... where one core may wait for information from other cores which in turn are waiting for the first one • Not all software is designed to use multi-cores • ... so some of the additional cores would be idle • Increased heat generation • ... which could cause damage to other components						
6(c)	1 mark per bullet point, max 3 marks • DRAM requires to be refreshed/charged SRAM does not request a refresh • DRAM stores each bit as charge SRAM uses flip-flop to store each bit • DRAM is less expensive to manufacture SRAM is more expensive to manufacture • DRAM has slower access speeds SRAM has faster access times • DRAM has higher <u>storage/bit/data density</u> SRAM has lower storage/bit/data density • DRAM is used in main memory SRAM is used in cache						

1(a)	1 mark for each correct row <table border="1"> <thead> <tr> <th data-bbox="363 185 608 315">Incorrect statement number</th><th data-bbox="608 185 1385 315">Corrected statement</th></tr> </thead> <tbody> <tr> <td data-bbox="363 315 608 416">1</td><td data-bbox="608 315 1385 416">The Program Counter (PC) stores the address of the next instruction to be fetched from memory.</td></tr> <tr> <td data-bbox="363 416 608 517">3</td><td data-bbox="608 416 1385 517">The Control Unit (CU) sends signals to other components on the control bus.</td></tr> <tr> <td data-bbox="363 517 608 678">4</td><td data-bbox="608 517 1385 678">The Memory Data Register (MDR) holds data to be stored in the memory address in the MAR. // The Memory Data Register (MDR) holds data read from the memory address in the MAR</td></tr> </tbody> </table>	Incorrect statement number	Corrected statement	1	The Program Counter (PC) stores the address of the next instruction to be fetched from memory.	3	The Control Unit (CU) sends signals to other components on the control bus.	4	The Memory Data Register (MDR) holds data to be stored in the memory address in the MAR. // The Memory Data Register (MDR) holds data read from the memory address in the MAR
Incorrect statement number	Corrected statement								
1	The Program Counter (PC) stores the address of the next instruction to be fetched from memory.								
3	The Control Unit (CU) sends signals to other components on the control bus.								
4	The Memory Data Register (MDR) holds data to be stored in the memory address in the MAR. // The Memory Data Register (MDR) holds data read from the memory address in the MAR								
1(b)	1 mark for name, 1 mark for corresponding role e.g. • Current Instruction Register // CIR • To store the instruction to be decoded / executed next • Status register // SR • To contain bits that can be referenced individually to indicate a state or event • Interrupt register • To store details of any interrupts that have occurred								
1(c)	1 mark each to max 4 • At the start / end of FE cycle the interrupt register is checked • The priority of any interrupts waiting is checked • if the priority of the interrupt is higher than the current process • the contents of the registers are stored on the stack • The relevant Interrupt Service Routine (ISR) / interrupt handler is called to process the interrupt • When the ISR has finished, a further check is made for higher priority interrupts • if no more interrupts of higher priority, the register contents are restored and the next FE cycle continues								

2(a)(i)	1 mark each to max 2 - max 1 for each set of marks • The control unit synchronises the actions of the processor • by sending a command / signal on each timing signal produced by the system clock • using / along the control bus
2(a)(ii)	1 mark each • $MAR \leftarrow [PC]$ and $PC \leftarrow [PC] + 1$ • $MDR \leftarrow [[MAR]]$ • $CIR \leftarrow [MDR]$ • Correct order • For example: • $MAR \leftarrow [PC]$ • $PC \leftarrow [PC] + 1$ • $MDR \leftarrow [[MAR]]$ • $CIR \leftarrow [MDR]$
2(b)	1 mark each • Using cache memory improves system performance • because cache is fast access memory close to the CPU • which stores frequently used instructions / data • so that they can be accessed faster than from RAM
2(c)	1 mark each to max 2 e.g. • HDMI transfers both audio and video using a single cable • HDMI has a high bandwidth • Data is transmitted in a stream • of uncompressed digital signals • HDMI uses a technology called Transition-Minimized Differential Signalling (TMDS)

3(a)	1 mark for system clock and 1 mark for Control Unit System clock • To synchronise operations • ... by creating and transmitting timing signals on the control bus Control Unit • Sends/receives control signals along control bus • Reads an instruction from the contents of the memory location whose address is stored in PC • Coordinates/synchronises the activity of other components in the CPU • Manages the execution of instructions • Controls communication between the components in the CPU
3(b)(i)	1 mark for the feature, 1 mark max for the matching reason e.g. Feature: clock speed • Higher clock speed means that more F-E cycles are executed per second // Higher clock speed results in more throughput Feature: bus width • Larger bus width means that more data transferred at the same time

3(b)(ii)	1 mark for correct port. 2 marks for explanation Port: USB / Universal Serial Bus Explanation: • A voltage change occurs when the drive is plugged in • The computer detects this voltage change • The code of the device is transferred to computer • ... the OS finds the code of the device in the list of devices • ... and loads the appropriate device driver
3(c)	1 mark for each bullet point (max 2) e.g. • DRAM requires constant refresh cycles unlike SRAM • DRAM has lower access speed than SRAM
3(d)(i)	1 mark for each bullet point (max 4) • The disc is spun at high speed • A laser is shone onto the disc to read / write • ... using optical head to move it into position • ... it follows the spiral track from the centre outwards • When writing the laser burns pits to represent the data • When reading the laser reflects from pits and lands • The reflection from a pit and a land is different • ... the differences are interpreted as 1 or 0
3(d)(ii)	1 mark for each bullet point (max 3) • The computer and optical disc reader/writer send and receive at different speeds • A buffer allows temporary storage of the data • ... so that the computer can transfer data to the buffer at the higher speed • ... and is not held up waiting for data to transfer // so the computer can carry on with other tasks • ... so the optical disc reader/writer is not overloaded • ... and so that data is transferred to optical disc reader/writer from the buffer at the slower rate
8(a)	1 mark for each bullet point (max 2) • To store the value of flags/bits • ... that can be changed/set/cleared after arithmetic / logical operations • To allow flags to be checked • ... to change instruction sequence
8(b)	1 mark for each bullet point (max 2) • Special purpose registers have a specified role in the machine whereas general purpose registers can be used for all purposes defined by the programmer • Special purpose registers hold the state of the program's execution while general purpose registers hold the program's data during operations

3(a)	1 mark for each correct purpose <table border="1" data-bbox="368 185 1323 678"> <thead> <tr> <th data-bbox="368 185 679 248">Register</th><th data-bbox="679 185 1323 248">Purpose</th></tr> </thead> <tbody> <tr> <td data-bbox="368 248 679 349">Program Counter (PC)</td><td data-bbox="679 248 1323 349">Stores the address of the next instruction to be fetched/executed</td></tr> <tr> <td data-bbox="368 349 679 450">Memory Address Register (MAR)</td><td data-bbox="679 349 1323 450">Stores the address of the memory location where data will be read from/written to</td></tr> <tr> <td data-bbox="368 450 679 577">Memory Data Register (MDR)</td><td data-bbox="679 450 1323 577">Stores the data read from the address in the MAR // stores the data to be written to the address in the MAR</td></tr> <tr> <td data-bbox="368 577 679 678">Index Register (IX)</td><td data-bbox="679 577 1323 678">Stores a number that will be added to the operand, to form the address of the data</td></tr> </tbody> </table>	Register	Purpose	Program Counter (PC)	Stores the address of the next instruction to be fetched/executed	Memory Address Register (MAR)	Stores the address of the memory location where data will be read from/written to	Memory Data Register (MDR)	Stores the data read from the address in the MAR // stores the data to be written to the address in the MAR	Index Register (IX)	Stores a number that will be added to the operand, to form the address of the data
Register	Purpose										
Program Counter (PC)	Stores the address of the next instruction to be fetched/executed										
Memory Address Register (MAR)	Stores the address of the memory location where data will be read from/written to										
Memory Data Register (MDR)	Stores the data read from the address in the MAR // stores the data to be written to the address in the MAR										
Index Register (IX)	Stores a number that will be added to the operand, to form the address of the data										
3(b)	1 mark for each bullet point (max 4) e.g. • HDMI has faster transfer rates than VGA • ... needed due to high resolution / large number of pixels of monitor // HDMI supports the high resolution of the monitor • HDMI supports video and audio transfer between computer and monitor speakers • ... so no separate sound cable is needed unlike VGA • HDMI is digital interface therefore no data is lost in transfer to analogue and back • HDMI is less prone to error/crosstalk/external interference										
3(c)	1 mark for each bullet point (max 4) e.g. • It manages the scheduling of processes // Decides which process is to be run next • Allows multi-tasking/multi-processing • Ensures fair access • Handles interrupts • Manages / allocates which resources the processes require • Facilitates the sharing and exchange of data between processes • Prevents interference between processes // conflict resolution										